# Objectifs de l'activité :

Théoriquement :

Étudier l'installation et l'architecture de Shinken (outil de supervision).

Pratiquement

* + Installer Shinken
  + Installer le module webui de Shinken
  + Dépanner les problèmes d'installation de Shinken
  + Étudier l'architecture de Shinken
    - Processus
    - Fichiers de configuration
    - Administration WEB

Matériel nécessaire

Une machine virtuelle avec Linux Debian 7.8 64 bits graphique installé pour avoir le navigateur et gedit. Il est préférable de cloner une MV existante.

La machine doit disposer d'un accès Internet (dans notre cas il faut la "bridger").

## Organisation

Travail individuel.

## Travail à faire

Chaque partie comporte des travaux à faire.

Un compte rendu démontrant que l'installation a été faite avec succès sera rédigé, les copies d'écran et les commentaires seront nécessaires. La dernière partie comporte des questions dont les réponses devront figurer sur le compte rendu.

## Auteur

Roger Sanchez (CERTA)

# Préambule : personnalisation de la machine Linux

## Modification du nom de la machine

Cette activité nécessite une machine virtuelle Linux Debian Wheezy qu'il faudra copier dans votre environnement et personnaliser.

On va modifier le nom de la machine en mettant un nom plus explicite mais pour éviter les doublons on le préfixera par nos initiales (modification du fichier **/etc/hostname** et de **/etc/hosts**, lancement du script **hostname.sh** et fermeture et ouverture du terminal.

echo rsShinken > /etc/hostname

cat /etc/hosts

127.0.0.1 localhost

127.0.1.1 debianWheezy.sisr5.org debianWheezy

/etc/init.d/hostname.sh

Attention à bien remplacer dans la commande précédente **debianWheezy** par vos valeurs (initiales + shinken 🡺 exemple **rsShinken**).

# Partie 1 : Installation de Shinken

Travail à faire 🡺 installer Shinken en suivant une procédure

Cette première étape installe les programmes nécessaires au fonctionnement de Shinken.

Shinken peut s'installer sur n'importe quel système disposant d'un interpréteur Python, y compris Windows ou Androïd.

On rappelle que l’on travaille ici avec une machine Linux Debian Wheezy.

Il faut impérativement utiliser une distribution Linux Debian 64 bits. On peut vérifier cela ainsi.

root@debianWheezy:~# uname -a

Linux debianWheezy 3.2.0-4-amd64 #1 SMP Debian 3.2.65-1+deb7u1 x86\_64 GNU/Linux

La commande **uname -a** affiche toutes les informations disponibles sur la version du système d'exploitation et le type de la machine.

Si on fait une recherche sur le paquet 3.2.0-4-amd64, on arrive sur le site <http://packages.debian.org/fr/wheezy/linux-image-3.2.0-4-amd64>, qui nous confirme qu'il s'agit d'une image 64 bits.

Distribution Debian

root@debianWheezy:~# cat /etc/debian\_version

7.8

On va maintenant appliquer la procédure décrite sur le site (avec quelques ajouts cependant) 🡺

[**http://shinken.readthedocs.org/en/latest/02\_gettingstarted/installations/shinken-installation.html**](http://shinken.readthedocs.org/en/latest/02_gettingstarted/installations/shinken-installation.html)

On choisit la méthode basée sur des "packages" pip.

La suite de commandes est la suivante :

root@debianWheezy:~# apt-get install python-pip python-pycurl

Lecture des listes de paquets... Fait

Construction de l'arbre des dépendances

Lecture des informations d'état... Fait

python-pycurl est déjà la plus récente version disponible.

python-pip est déjà la plus récente version disponible.

Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :

libmozjs17d xulrunner-17.0

Veuillez utiliser « apt-get autoremove » pour les supprimer.

0 mis à jour, 0 nouvellement installés, 0 à enlever et 0 non mis à jour.

root@debianWheezy:~# adduser shinken

Ajout de l'utilisateur « shinken » ...

Ajout du nouveau groupe « shinken » (1001) ...

Ajout du nouvel utilisateur « shinken » (1001) avec le groupe « shinken » ...

Création du répertoire personnel « /home/shinken »...

Copie des fichiers depuis « /etc/skel »...

Entrez le nouveau mot de passe UNIX :

Retapez le nouveau mot de passe UNIX :

passwd : le mot de passe a été mis à jour avec succès

Modification des informations relatives à l'utilisateur shinken

Entrez la nouvelle valeur ou « Entrée » pour conserver la valeur proposée

Nom complet []:

N° de bureau []:

Téléphone professionnel []:

Téléphone personnel []:

Autre []:

Cette information est-elle correcte ? [O/n]

root@debianWheezy:~# pip install shinken

Downloading/unpacking shinken

Downloading Shinken-2.2.tar.gz (27.1Mb): 27.1Mb downloaded

Running setup.py egg\_info for package shinken

Parser error no such option: --egg-base

('updating path in %s', 'build/shinken.cfg')

warning: no files found matching '.reg' under directory 'windows'

warning: no files found matching 'test'

warning: no files found matching '\*.txt'

warning: no previously-included files matching '.git' found anywhere in distribution

warning: no previously-included files matching '\*.pyc' found anywhere in distribution

warning: no previously-included files matching '\*.pyo' found anywhere in distribution

warning: no previously-included files matching '.debug' found anywhere in distribution

warning: no previously-included files matching '.pid' found anywhere in distribution

warning: no previously-included files matching '.exe' found anywhere in distribution

warning: no previously-included files matching '\*~' found anywhere in distribution

warning: no files found matching 'var/void\_for\_git'

Notice: for better performances for the daemons communication, you should install the python-cherrypy3 lib

Shinken setup done

Installing collected packages: shinken

Running setup.py install for shinken

('updating path in %s', 'build/shinken.cfg')

warning: no files found matching '.reg' under directory 'windows'

warning: no files found matching 'test'

warning: no files found matching '\*.txt'

warning: no previously-included files matching '.git' found anywhere in distribution

warning: no previously-included files matching '\*.pyc' found anywhere in distribution

warning: no previously-included files matching '\*.pyo' found anywhere in distribution

warning: no previously-included files matching '.debug' found anywhere in distribution

warning: no previously-included files matching '.pid' found anywhere in distribution

warning: no previously-included files matching '.exe' found anywhere in distribution

warning: no previously-included files matching '\*~' found anywhere in distribution

warning: no files found matching 'var/void\_for\_git'

Changing owner of /etc/shinken to shinken:shinken

Changing owner of /var/run/shinken to shinken:shinken

Changing owner of /var/log/shinken to shinken:shinken

Changing owner of /var/lib/shinken/ to shinken:shinken

Changing owner of /var/lib/shinken/libexec to shinken:shinken

Changing owner of /usr/bin/shinken-broker to shinken:shinken

Changing owner of /usr/bin/shinken-arbiter to shinken:shinken

Changing owner of /usr/bin/shinken-reactionner to shinken:shinken

Changing owner of /usr/bin/shinken-discovery to shinken:shinken

Changing owner of /usr/bin/shinken-scheduler to shinken:shinken

Changing owner of /usr/bin/shinken-poller to shinken:shinken

Changing owner of /usr/bin/shinken to shinken:shinken

Changing owner of /usr/bin/shinken-receiver to shinken:shinken

Notice: for better performances for the daemons communication, you should install the python-cherrypy3 lib

Shinken setup done

Successfully installed shinken

Cleaning up...

S'il n'y a pas eu de problème, on doit pouvoir démarrer shinken.

root@debianWheezy:~# service shinken start

Starting scheduler:

. ok

Starting poller:

. ok

Starting reactionner:

. ok

Starting broker:

. ok

Starting receiver:

. ok

Starting arbiter:

. ok

Les modules présents après l'installation sont les modules obligatoires :

root@debianWheezy:~# ls /var/lib/shinken/modules/

dummy\_arbiter dummy\_broker\_external dummy\_scheduler

dummy\_broker dummy\_poller \_\_init\_\_.py

Dans le répertoire de configuration des modules optionnels on ne trouve qu'un exemple :

root@debianWheezy:~# ls /etc/shinken/modules/

sample.cfg

A ce stade shinken est opérationnel mais il nous manque un outil pour visualiser notre supervision.

# Partie 2 : Installation de l'interface webui

Shinken comprend une interface web (webui) qu'il faut installer. Elle s'appuie sur un serveur http interne ne nécessitant pas de composant supplémentaire comme apache par exemple.

Il faut systématiquement se connecter avec l'utilisateur shinken dès lors qu'on ajoute des éléments à shinken. Car les différents exécutables de shinken utilisent l'uid shinken. Il faut donc que les nouveaux fichiers installés bénéficient des droits shinken et non root.

On va donc ouvrir un nouveau terminal et se connecter avec l'utilisateur shinken

root@debianWheezy:~# su - shinken

shinken@debianWheezy:~$

On trouvera ici une explication succincte de l'outil client shinken :

<http://www.shinken-monitoring.org/wiki/shinken.io/start>

L'outil client shinken permet de rechercher et d'installer facilement des modules et des "packs" shinken à partir de l'url <http://shinken.io>

Il faut tout d'abord initialiser l'outil client shinken.

shinken@debianWheezy:~$ shinken --init

Creating ini section paths

Creating ini section shinken.io

Saving the new configuration file /home/shinken/.shinken.ini

Ce qui crée le fichier shinken.ini dans le répertoire /home/shinken :

[paths]

# # set the paths according to your setup. defaults follow

# the linux standard base

etc = /etc/shinken

lib = /var/lib/shinken

share = %(lib)s/share

cli = %(lib)s/cli

packs = %(etc)s/packs

modules = %(lib)s/modules

doc = %(lib)s/doc

inventory = %(lib)s/inventory

libexec = %(lib)s/libexec

[shinken.io]

# options for connection to the shinken.io website.

# proxy: curl style, should look as http://user:password@server:3128

# api\_key: useful for publishing packages or earn xp after each install. create an account at http://shinken.io and go to http://shinken.io/~

proxy =

api\_key =

On va maintenant installer l'interface web.

shinken@debianWheezy:~$ shinken install webui

Grabbing : webui

OK webui

Après cela on trouve un nouveau module.

root@debianWheezy:~# ls /var/lib/shinken/modules/

dummy\_arbiter dummy\_broker\_external dummy\_scheduler webui

dummy\_broker dummy\_poller \_\_init\_\_.py

Et un nouveau fichier de configuration.

root@debianWheezy:~# ls /etc/shinken/modules/

sample.cfg webui.cfg

Dont le contenu est le suivant :

## Module: webui

## Loaded by: Broker

# The Shinken web interface and integrated web server.

define module {

module\_name webui

module\_type webui

host 0.0.0.0 ; All interfaces = 0.0.0.0

port 7767

auth\_secret CHANGE\_ME ; CHANGE THIS or someone could forge cookies

allow\_html\_output 1 ; Allow or not HTML chars in plugins output.

; WARNING: Allowing can be a security issue.

max\_output\_length 1024 ; Maximum output length for plugin output in webui

manage\_acl 1 ; Use contacts ACL. 0 allow actions for all.

play\_sound 0 ; Play sound on new non-acknowledged problems.

login\_text Welcome on Shinken WebUI ; Text in the login form.

## Modules for WebUI

# - auth-htpasswd = Use an htpasswd file for auth backend.

# - auth-active-directory = Use AD for auth backend (and retrieve photos).

# - auth-cfg-password = Use the password setted in Shinken contact for auth.

# - ui-pnp = Use PNP graphs in the UI.

# - ui-graphite = Use graphs from Graphite time series database.

# - mongodb = Save user preferences to a Mongodb database

# - SQLitedb = Save user preferences to a SQLite database

modules

## Advanced Options

# Don't play with them on your production server ;)

#http\_backend auto ; Choice is: auto, wsgiref or cherrypy if available

#remote\_user\_enable 1 ; If WebUI is behind a web server which

; has already authentified user, enable.

#remote\_user\_enable 2 ; Look for remote user in the WSGI environment

; instead of the HTTP header. This allows

; for fastcgi (flup) and scgi (flupscgi)

; integration, eg. with the apache modules.

#remote\_user\_variable X\_Remote\_User ; Set to the HTTP header containing

; the authentificated user s name, which

; must be a Shinken contact.

# If you got external plugins (pages) to load on webui

#additional\_plugins\_dir

}

Pour le prendre en compte il faut relancer shinken.

root@debianWheezy:~# service shinken restart

Restarting scheduler

. ok

Restarting poller

. ok

Restarting reactionner

. ok

Restarting broker

. ok

Restarting receiver

. ok

Restarting arbiter

Doing config check

. ok

. ok

Peut-on se connecter à webui ?



On ne peut pas se connecter car il n'y a pas de liaison entre le module Broker et webui.

Il faut modifier le fichier de configuration du broker /etc/shinken/brokers/broker-master.cfg ainsi (en lui ajoutant la ligne ***modules webui***)

#===============================================================================

# BROKER (S1\_Broker)

#===============================================================================

# Description: The broker is responsible for:

# - Exporting centralized logs of all Shinken daemon processes

# - Exporting status data

# - Exporting performance data

# - Exposing Shinken APIs:

# - Status data

# - Performance data

# - Configuration data

# - Command interface

# https://shinken.readthedocs.org/en/latest/08\_configobjects/broker.html

#===============================================================================

define broker {

broker\_name broker-master

address localhost

port 7772

spare 0

## Optional

manage\_arbiters 1 ; Take data from Arbiter. There should be only one

; broker for the arbiter.

manage\_sub\_realms 1 ; Does it take jobs from schedulers of sub-Realms?

timeout 3 ; Ping timeout

data\_timeout 120 ; Data send timeout

max\_check\_attempts 3 ; If ping fails N or more, then the node is dead

check\_interval 60 ; Ping node every N seconds

## Modules

# Default: None

# Interesting modules that can be used:

# - simple-log = just all logs into one file

# - livestatus = livestatus listener

# - tondodb-mysql = NDO DB support

# - npcdmod = Use the PNP addon

# - graphite = Use a Graphite time series DB for perfdata

# - webui = Shinken Web interface

# - glpidb = Save data in GLPI MySQL database

modules webui

# Enable https or not

use\_ssl 0

# enable certificate/hostname check, will avoid man in the middle attacks

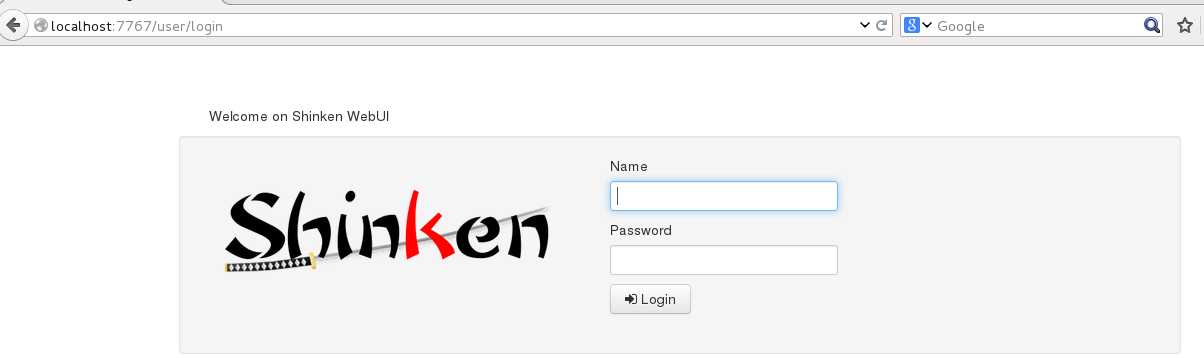
hard\_ssl\_name\_check 0

## Advanced

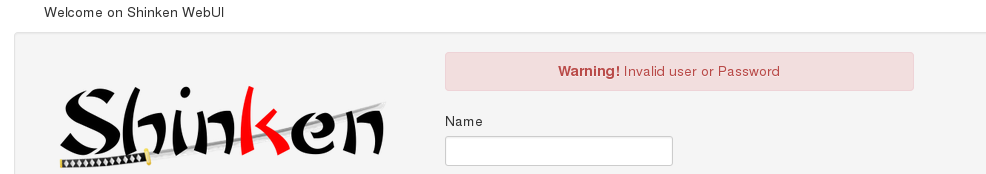
realm All

}

On redémarre shinken et on essaie à nouveau :



On saisit le couple admin/admin comme indiqué dans la documentation mais…



Il faut implémenter une méthode d'authentification pour **webui** (voir <http://shinken.readthedocs.org/en/latest/11_integration/webui.html?highlight=webui>)

Nous allons utiliser une méthode simple basée sur les contacts, c’est-à-dire sur les utilisateurs déclarés dans /etc/shinken/contacts/admin.cfg

# This is a default admin

# CHANGE ITS PASSWORD!

define contact{

use generic-contact

contact\_name admin

email shinken@localhost

pager 0600000000 ; contact phone number

password admin

is\_admin 1

expert 1

}

On va encore un fois utiliser le client shinken en recherchant les modules d'authentification associés à webui.

shinken@debianWheezy:~$ shinken search webui auth

auth-active-directory (naparuba) [module,broker,webui,auth,authentification,auth-active-directory] : Active directory auth for WebUI

auth-cfg-password (naparuba) [module,auth,authentification,mod-auth-cfg-password,auth-cfg-password,cfg-password,webui] : Shinken module for UI authentification from simple password for configuration file

auth-htpasswd (naparuba) [module,webui,auth,authentification] : Shinken module for UI authentification from Apache passwd files

auth-pam (mingbo\_wan) [module,auth,authentification,auth-cfg-pam,cfg-pam,webui] : Shinken module for UI authentification via pam

auth-ws-glpi (mohierf) [module,broker,webui,glpi,auth,authentication] : Authenticate Web UI users upon a Glpi Web Service login.

On installe le bon module :

shinken@debianWheezy:~$ shinken install auth-cfg-password

Grabbing : auth-cfg-password

OK auth-cfg-password

On vérifie l'installation :

root@debianWheezy:~# ls /var/lib/shinken/modules/

auth-cfg-password dummy\_broker dummy\_poller \_\_init\_\_.py

dummy\_arbiter dummy\_broker\_external dummy\_scheduler webui

root@debianWheezy:~# ls /etc/shinken/modules/

auth\_cfg\_password.cfg sample.cfg webui.cfg

On regarde le fichier de configuration du nouveau module :

## Module: auth-cfg-password

## Loaded by: WebUI

# Check authentification using password parameter in contact definition.

define module {

module\_name auth-cfg-password

module\_type cfg\_password\_webui

}

Il faut modifier le fichier de configuration du module webui pour lui indiquer la méthode d'authentification.

Modification dans cfg de webui

port 7767

auth\_secret CHANGE\_ME ; CHANGE THIS or someone could forge cookies

allow\_html\_output 1 ; Allow or not HTML chars in plugins output.

; WARNING: Allowing can be a security issue.

max\_output\_length 1024 ; Maximum output length for plugin output in webui

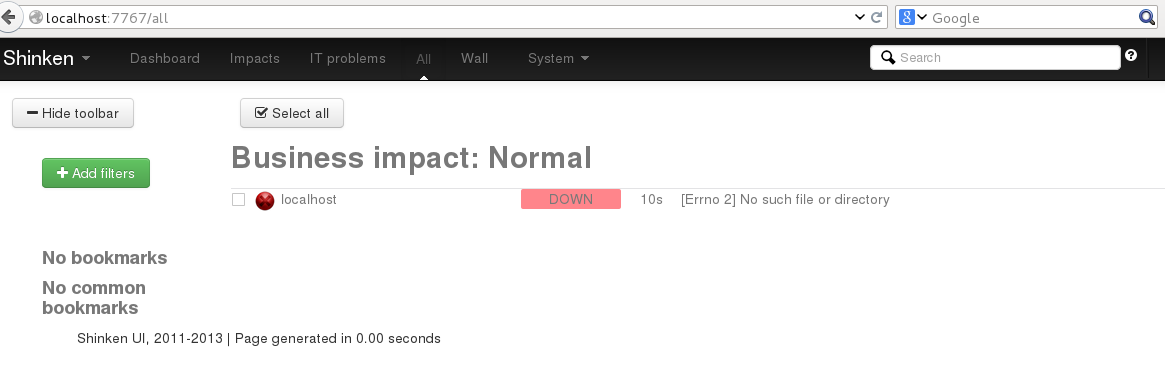
manage\_acl 1 ; Use contacts ACL. 0 allow actions for all.

play\_sound 0 ; Play sound on new non-acknowledged problems.

login\_text Welcome on Shinken WebUI ; Text in the login form.

modules auth-cfg-password

Enfin on redémarre shinken et on se connecte de nouveau :



On a maintenant accès à l'interface même si pour l'instant, ce n'est pas très probant, mais on verra ça plus tard.

L'interface principale webUI est accessible par le port **7767**. D'autres interfaces sont utilisables. Le module **broker** dont dépend webui permet une entière liberté de choix.



On saisit **admin/admin.**

Shinken permet de gérer des utilisateurs (contacts). Ils sont définis dans le répertoire suivant :

root@debianWheezy:~# ls /etc/shinken/contacts

admin.cfg guest.cfg

Voici ce qu'on y trouve après l'installation :

root@debianWheezy:~# cat /etc/shinken/contacts/admin.cfg

# This is a default admin

# CHANGE ITS PASSWORD!

Le mot clé **define** permet de créer une instance d'objet. Ici un objet contact.

define contact{

use generic-contact

contact\_name admin

email shinken@localhost

pager 0600000000 ; contact phone number

password admin

is\_admin 1

expert 1

}

root@debianWheezy:~# cat /etc/shinken/contacts/guest.cfg

# This is a default guest user

# CHANGE ITS PASSWORD or remove it

define contact{

use generic-contact

contact\_name guest

email guest@localhost

password guest

can\_submit\_commands 0

}

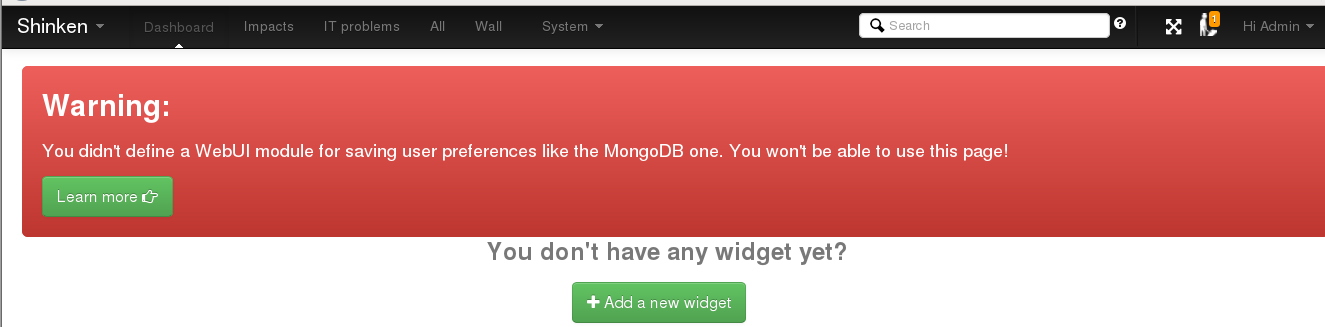
Par défaut, Shinken configure 2 contacts :

* **admin** : un compte avec le statut d'administrateur (🡺**is\_admin 1**)
* **guest :** un utilisateur avec le statut d'invité. Il peut consulter l'état des différents équipements surveillés mais ne peut effectuer aucune action (🡺**can\_submit\_commands 0**)

Il faudrait bien sûr changer ce mot de passe, on ne le fera pas pour l'instant.

Il nous reste quelques réglages à opérer pour avoir une installation prête pour la supervision.

En effet, le **dashboard** (tableau de bord) ne fonctionne pas.



Comme cela nous est dit, li faut installer une base de données pour stocker les préférences utilisateurs. On nous propose d'ailleurs d'installer mongodb (on pourrait choisir sqlLITE).

Pour installer mongodb il faut procéder en 2 étapes.

Installer les parquets :

root@debianWheezy:~# apt-get install mongodb python-pymongo

Lecture des listes de paquets... Fait

Construction de l'arbre des dépendances

Lecture des informations d'état... Fait

….

Fait.

[ ok ] Starting database: mongodb.

Paramétrage de mongodb-dev (1:2.0.6-1.1) ...

Paramétrage de mongodb (1:2.0.6-1.1) ...

Paramétrage de python-bson (2.2-4+deb7u1) ...

Paramétrage de python-bson-ext (2.2-4+deb7u1) ...

Paramétrage de python-pymongo (2.2-4+deb7u1) ...

Paramétrage de python-gridfs (2.2-4+deb7u1) ...

Paramétrage de python-pymongo-ext (2.2-4+deb7u1

Installer le module mongodb pour shinken (rappel : terminal avec su – shinken toujours).

shinken@debianWheezy:~$ shinken install mod-mongodb

Grabbing : mod-mongodb

OK mod-mongodb

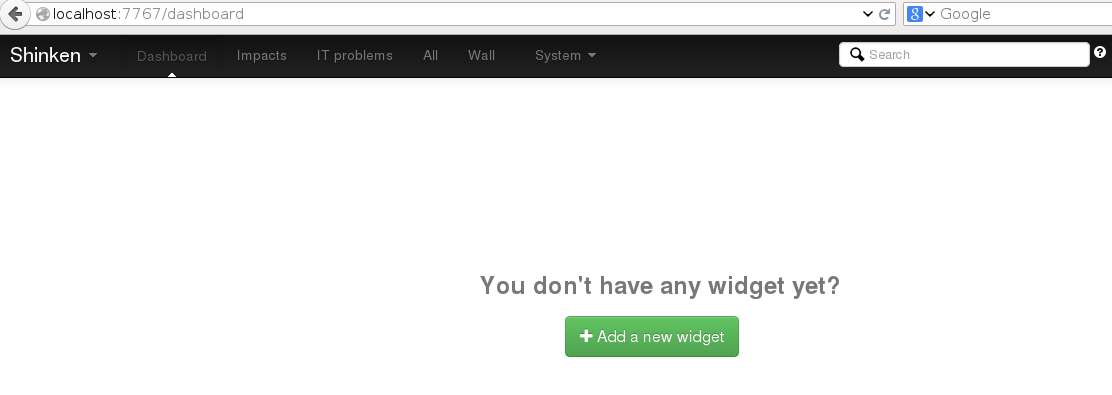
Puis enfin établir une relation entre webui et mongodb dans le fichier de configuration de webui /etc/shinken/modules/webui.cfg :

modules auth-cfg-password,mongodb

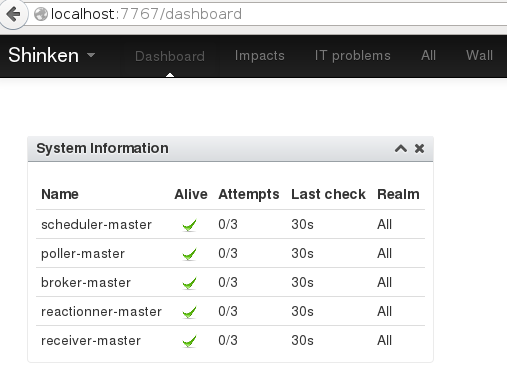
(Attention pas d'espace entre les différents modules).

Pour finir, il ne faut pas oublier de relancer shinken

On doit obtenir au final ceci :



L'installation de "**widget**" (composant d'interface graphique permettant d'obtenir des informations) se fait sans problème, par exemple on peut installer le widget **"Sytem state".**



**Dernière remarque** : l'outil shinken nous permet de voir ce qu'on a installé grâce à l'option **inventory :**

shinken@debianWheezy:~$ shinken inventory

auth-cfg-password

webui

# Partie 3 : Architecture de Shinken

Travail à faire 🡺 observer le contenu des différents répertoires de Shinken et exécuter les commandes proposées.

A ce stade, on a installé les modules obligatoires de **shinken** et les modules optionnels **webui** et **mongodb**.

La version stable de Shinken présentée ici est la 2.2.

**Répertoires associés à Shinken**

root@debianWheezy:~# find / -name shinken

/var/lib/shinken

/var/lib/shinken/doc/build/html/\_modules/shinken

/var/log/shinken

/var/mail/shinken

/run/shinken

/run/console/shinken

/home/shinken

/usr/local/lib/python2.7/dist-packages/shinken

/usr/bin/shinken

/etc/init.d/shinken

/etc/default/shinken

/etc/shinken

root@debianWheezy:~# ls /var/lib/shinken/modules/

auth-cfg-password dummy\_broker dummy\_poller \_\_init\_\_.py

dummy\_arbiter dummy\_broker\_external dummy\_scheduler webui

root@debianWheezy:~# ls /var/lib/shinken/libexec

check\_netint.pl external\_mapping.py notify\_by\_xmpp.ini

check\_shinken.py link\_libvirt\_host\_vm.py notify\_by\_xmpp.py

discovery link\_vmware\_host\_vm.py send\_nsca.py

dump\_vmware\_hosts.py link\_xen\_host\_vm.py service\_dependency\_mapping.py

* Répertoire contenant les modules : **/var/lib/shinken/modules.**
* Répertoire contenant les sondes : **/var/lib/shinken/libexec.**
* Fichiers de configuration : **/etc/shinken.**
* Lanceurs : **/etc/init.d/shinken.**
* Démons shinken : **/usr/bin.**
* Fichiers de log : **/var/log/shinken.**

Les fichiers/répertoires de configuration sont :

root@debianWheezy:~# ls /etc/shinken/

arbiters daemons hosts realms servicegroups

brokers dependencies modules receivers services

certs dev.cfg notificationways resource.d shinken.cfg

commands discovery packs sample templates

contactgroups escalations pollers sample.cfg timeperiods

contacts hostgroups reactionners schedulers

Quand on démarre **shinken**, on constate qu'en réalité le script démarre plusieurs démons.

Chacun de ces démons dispose d'un fichier de configuration dans /etc/shinken.

root@rsShinken:/usr/local/shinken/etc# service shinken start

Starting scheduler:

ok

Starting poller:

. ok

Starting reactionner:

. ok

Starting broker:

. ok

Starting receiver:

. ok

Starting arbiter:

. ok

C'est ici la principale différence entre **shinken** et son aîné **nagios**. Son architecture est modulaire, elle repose sur le principe Unix : à une tâche un outil.

**Shinken** utilise six processus (ou composants) différents qui travaillent ensemble et permettent d'obtenir une flexibilité bien supérieure au **nagios** originel. Surtout, contrairement à un processus monolithique, cette architecture favorise la montée en charge.

**Shinken** se décompose ainsi :

* **L’arbitre (Arbiter) :** il est notamment responsable de la validation et du chargement de la configuration, de la découpe en différentes parties, et de l’envoi aux autres éléments.
* **L’ordonnanceur (Scheduler) :** il est chargé d’ordonnancer les « checks », d’analyser leurs résultats et de déclencher une action en fonction de ces derniers si c’est nécessaire. Ce n’est pas lui qui lance les checks ou les notifications, il ne fait que rediriger les informations. Il garde juste dans une file d’attente les « checks » en attentes (*pending*) et notification pour les autres éléments (collecteurs ou « Réactionneurs »). Il peut y avoir plusieurs ordonnanceurs, c'est d'ailleurs conseillé.
* **Le collecteur (Poller) :** son rôle est de lancer les plugins en fonction des requêtes des ordonnanceurs. Ces plugins, qui peuvent être ceux de Nagios, vont aller interroger le système surveillé et retourner un résultat indiquant l'état. Lorsqu’un plugin renvoie un résultat, il le transmet à l’ordonnanceur. Il peut y avoir plusieurs collecteurs.
* **Le « réactionneur » (Reactionner) :** il est chargé de l'envoi des notifications et de lancer les « event\_handlers » (action automatique programmable).
* **Le « courtier » (Broker) :** son rôle est de prendre des données sur les schedulers (comme les statuts par exemple) et de les rendre disponibles à l'externe de Shinken.
* **Le « receveur » (Receiver) :** son rôle est de recevoir les données d'acquisition passive et de les acheminer vers le bon scheduler responsable de faire la corrélation et le traitement des statuts. Il peut y avoir plusieurs receveurs.

On peut observer les processus en cours d'exécution grâce à la commande suivante :

root@debianWheezy:~# ps -AHF | grep shinken

root 11718 11464 0 14031 1616 0 12:24 pts/1 00:00:00 su - shinken

shinken 11726 11718 0 5191 3588 0 12:24 pts/1 00:00:00 -su

root 10138 24543 0 1963 888 0 15:18 pts/0 00:00:00 grep shinken

shinken 30847 1 1 58385 17324 0 12:58 ? 00:02:22 python2.7 /usr/bin/shinken-scheduler -d -c /etc/shinken/daemons/schedulerd.ini

shinken 30849 30847 0 28926 12924 0 12:58 ? 00:00:00 python2.7 /usr/bin/shinken-scheduler -d -c /etc/shinken/daemons/schedulerd.ini

shinken 30922 1 0 56134 16936 0 12:58 ? 00:01:18 python2.7 /usr/bin/shinken-poller -d -c /etc/shinken/daemons/pollerd.ini

shinken 30925 30922 0 32398 11112 0 12:58 ? 00:00:04 python2.7 /usr/bin/shinken-poller -d -c /etc/shinken/daemons/pollerd.ini

shinken 31408 30922 0 39709 15756 0 12:59 ? 00:00:02 python2.7 /usr/bin/shinken-poller -d -c /etc/shinken/daemons/pollerd.ini

shinken 30999 1 0 56135 16968 0 12:58 ? 00:01:10 python2.7 /usr/bin/shinken-reactionner -d -c /etc/shinken/daemons/reactionnerd.ini

shinken 31002 30999 0 32398 11072 0 12:58 ? 00:00:04 python2.7 /usr/bin/shinken-reactionner -d -c /etc/shinken/daemons/reactionnerd.ini

shinken 31419 30999 0 39710 15792 0 12:59 ? 00:00:01 python2.7 /usr/bin/shinken-reactionner -d -c /etc/shinken/daemons/reactionnerd.ini

shinken 31143 1 1 74755 17692 0 12:59 ? 00:01:47 python2.7 /usr/bin/shinken-broker -d -c /etc/shinken/daemons/brokerd.ini

shinken 31145 31143 0 33214 14040 0 12:59 ? 00:00:04 python2.7 /usr/bin/shinken-broker -d -c /etc/shinken/daemons/brokerd.ini

shinken 31383 31143 0 40446 19292 0 12:59 ? 00:00:05 python2.7 /usr/bin/shinken-broker -d -c /etc/shinken/daemons/brokerd.ini

shinken 31222 1 0 39726 16044 0 12:59 ? 00:01:04 python2.7 /usr/bin/shinken-receiver -d -c /etc/shinken/daemons/receiverd.ini

shinken 31224 31222 0 28937 12876 0 12:59 ? 00:00:00 python2.7 /usr/bin/shinken-receiver -d -c /etc/shinken/daemons/receiverd.ini

shinken 31319 1 0 50465 18432 0 12:59 ? 00:01:07 python2.7 /usr/bin/shinken-arbiter -d -c /etc/shinken/shinken.cfg

shinken 31322 31319 0 29983 17284 0 12:59 ? 00:00:00 python2.7 /usr/bin/shinken-arbiter -d -c /etc/shinken/shinken.cfg

*L'option –A affiche tous les processus.*

*L'option –H affiche l'arborescence des processus.*

*L'option –F affiche toutes les informations associées au processus.*

On s’aperçoit ici que les processus Shinken sont en fait pris en charge par l’interpréteur Python.

On constate aussi, **élément important à noter**, que les processus ne sont pas exécutés avec les droits **root** mais avec les droits d'un utilisateur **shinken**.

On voit aussi les fichiers de paramètres utilisés par les commandes **(.ini).**

Tous les programmes Python utilisés par Shinken se trouvent dans **/usr/bin** et les fichiers de configuration dans **/etc/shinken**.

On peut repérer les ports TCP écoutés par Shinken en exécutant la commande suivante :

root@debianWheezy:~# netstat -tlnp | grep "LISTEN .\*python"

tcp 0 0 127.0.0.1:39026 0.0.0.0:\* LISTEN 30925/python2.7

tcp 0 0 127.0.0.1:47924 0.0.0.0:\* LISTEN 31145/python2.7

tcp 0 0 0.0.0.0:7767 0.0.0.0:\* LISTEN 31383/python2.7

tcp 0 0 127.0.0.1:45047 0.0.0.0:\* LISTEN 31322/python2.7

tcp 0 0 0.0.0.0:7768 0.0.0.0:\* LISTEN 30847/python2.7

tcp 0 0 0.0.0.0:7769 0.0.0.0:\* LISTEN 30999/python2.7

tcp 0 0 127.0.0.1:7770 0.0.0.0:\* LISTEN 31319/python2.7

tcp 0 0 0.0.0.0:7771 0.0.0.0:\* LISTEN 30922/python2.7

tcp 0 0 0.0.0.0:7772 0.0.0.0:\* LISTEN 31143/python2.7

tcp 0 0 0.0.0.0:7773 0.0.0.0:\* LISTEN 31222/python2.7

tcp 0 0 127.0.0.1:33502 0.0.0.0:\* LISTEN 31224/python2.7

tcp 0 0 127.0.0.1:50599 0.0.0.0:\* LISTEN 31002/python2.7

tcp 0 0 127.0.0.1:55719 0.0.0.0:\* LISTEN 30849/python2.7

***options de netstat*** *: t pour tcp, l pour listenning, n pour numeric et p pour program*

**Où sont définis ces différents ports ?**

Dans les fichiers de configuration associés à chaque module, par exemple pour le module **poller,** ontrouve dans **/etc/shinken/pollers/poller-master.cfg** une directive **port** :

#===============================================================================

# POLLER (S1\_Poller)

#===============================================================================

# Description: The poller is responsible for:

# - Active data acquisition

# - Local passive data acquisition

# https://shinken.readthedocs.org/en/latest/08\_configobjects/poller.html

#===============================================================================

define poller {

poller\_name poller-master

address localhost

port 7771

Chaque modification dans les fichiers de configuration nécessite le redémarrage de l'Arbiter**.**

Le démarrage de l**’arbiter** se fait :

- soit en relançant l’ensemble des démons avec /etc/init.d/shinken restart (ou bien la commande **service**),

- soit en lançant la commande suivante pour ne lancer que **l'arbiter.**

/etc/init.d/shinken-arbiter restart

Restarting arbiter

Doing config check

# Partie 4 : Base de connaissances

## Qu'a-t-on appris avec cette installation ? (à part savoir appliquer une procédure bien sûr)

* L'utilisation de la commande **pip.**
* L'utilisation du l'outil client **shinken.**
* L'installation de modules à partir de l'url <http://shinken.io> avec **shinken install.**
* Les principaux répertoires de shinken.
* L'architecture de shinken.
* Quelques exemples de configuration.
* Les commandes de gestion.

On trouvera un mémo Shinken très intéressant à cette adresse :

<http://www.premont.fr/html.php?page=replace.php&file=memos/Systems/Monitoring/memo_shinken>

Sur **pip** on peut consulter :

<http://philpep.org/blog/etat-de-l-art-du-packaging-python>

<http://apprendre-python.com/page-pip-installer-librairies-automatiquement>

## Documenter et dépanner l'installation/configuration

1. Où doivent se trouver les scripts gérés par la commande **service** ?
2. Que fait la commande suivante ?

/usr/bin/shinken-arbiter -d -c /etc/shinken/shinken.cfg

1. Quel est le port d'écoute TCP et quel est le processus à l'écoute ?

netstat -tlnp | grep "LISTEN .\*python"

tcp 0 0 0.0.0.0:7768 0.0.0.0:\* LISTEN 15657/python

1. Que fait la commande suivante ?

root@debianWheezy:~# service shinken check

# Correction :

1. Où doivent se trouver les scripts gérés par la commande service ?

Dans le répertoire **/etc/init.d**

1. Que fait la commande suivante ?

/usr/local/shinken/bin/shinken-arbiter -d -c /usr/local/shinken/etc/nagios.cfg -c /usr/local/shinken/etc/shinken-specific.cfg

Cette commande redémarre l'**arbiter** en utilisant les fichiers de configurations **shinken.cfg.**

L'option **–d** signifie que le processus sera exécuté comme un "démon".

On voit aussi ici l'ensemble des fichiers de configuration chargé au démarrage.

root@debianWheezy:~# /usr/bin/shinken-arbiter -d -c /etc/shinken/shinken.cfg

[1424356887] INFO: [Shinken] Shinken 2.2

[1424356887] INFO: [Shinken] Copyright (c) 2009-2014:

[1424356887] INFO: [Shinken] Gabes Jean (naparuba@gmail.com)

[1424356887] INFO: [Shinken] Gerhard Lausser, Gerhard.Lausser@consol.de

[1424356887] INFO: [Shinken] Gregory Starck, g.starck@gmail.com

[1424356887] INFO: [Shinken] Hartmut Goebel, h.goebel@goebel-consult.de

[1424356887] INFO: [Shinken] License: AGPL

[1424356887] INFO: [Shinken] Loading configuration

[1424356887] INFO: [Shinken] [config] opening '/etc/shinken/shinken.cfg' configuration file

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/notify-service-by-android-sms.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/check\_dig.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/check\_tcp.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/restart-shinken.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/detailled-service-by-email.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/notify-host-by-xmpp.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/check\_host\_alive.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/configuration-check.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/check\_snmp\_time.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/notify-host-by-android-sms.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/check\_nrpe\_args.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/notify-service-by-email.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/notify-service-by-xmpp.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/check\_nrpe.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/reload-shinken.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/notify-host-by-email.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/detailled-host-by-email.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/check\_snmp\_service.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/check\_ping.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/commands/check\_snmp\_storage.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/timeperiods/24x7.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/timeperiods/none.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/timeperiods/us-holidays.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/timeperiods/workhours.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/escalations/sample.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/dependencies/sample.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/templates/time\_templates.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/templates/generic-contact.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/templates/generic-host.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/templates/srv-pnp.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/templates/generic-service.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/notificationways/email.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/notificationways/detailled-email.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/notificationways/sms-android.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/servicegroups/sample.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/hostgroups/linux.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/contactgroups/users.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/contactgroups/admins.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/hosts/localhost.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/services/services.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/contacts/admin.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/contacts/guest.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/packs/readme.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/modules/webui.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/modules/auth\_cfg\_password.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/modules/sample.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/arbiters/arbiter-master.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/schedulers/scheduler-master.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/pollers/poller-master.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/reactionners/reactionner-master.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/reactionners/reactionner-android-sms.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/brokers/broker-master.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/receivers/receiver-master.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/realms/all.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/resource.d/paths.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/resource.d/active-directory.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/resource.d/nmap.cfg'

[1424356887] INFO: [Shinken] Processing object config file '/etc/shinken/resource.d/snmp.cfg'

[1424356887] INFO: [Shinken] And arbiter is launched with the hostname:debianWheezy from an arbiter point of view of addr:debianWheezy

[1424356887] INFO: [Shinken] I am the master Arbiter: arbiter-master

[1424356887] INFO: [Shinken] My own modules:

[1424356887] INFO: [Shinken] Modules directory: /var/lib/shinken/modules

[1424356887] INFO: [Shinken] Modules directory: /var/lib/shinken/modules

[1424356887] INFO: [Shinken] I correctly loaded the modules: []

[1424356887] INFO: [Shinken] All: (in/potential) (schedulers:1) (pollers:1/1) (reactionners:1/1) (brokers:1/1) (receivers:1/1)

[1424356887] INFO: [Shinken] Running pre-flight check on configuration data...

[1424356887] INFO: [Shinken] Checking global parameters...

[1424356887] INFO: [Shinken] Checking hosts...

[1424356887] INFO: [Shinken] Checked 1 hosts

[1424356887] INFO: [Shinken] Checking hostgroups...

[1424356887] INFO: [Shinken] Checked 1 hostgroups

[1424356887] INFO: [Shinken] Checking contacts...

[1424356887] INFO: [Shinken] Checked 2 contacts

[1424356887] INFO: [Shinken] Checking contactgroups...

[1424356887] INFO: [Shinken] Checked 2 contactgroups

[1424356887] INFO: [Shinken] Checking notificationways...

[1424356887] INFO: [Shinken] Checked 3 notificationways

[1424356887] INFO: [Shinken] Checking escalations...

[1424356887] INFO: [Shinken] Checked 0 escalations

[1424356887] INFO: [Shinken] Checking services...

[1424356887] INFO: [Shinken] Checked 0 services

[1424356887] INFO: [Shinken] Checking servicegroups...

[1424356887] INFO: [Shinken] Checked 0 servicegroups

[1424356887] INFO: [Shinken] Checking timeperiods...

[1424356887] INFO: [Shinken] Checked 4 timeperiods

[1424356887] INFO: [Shinken] Checking commands...

[1424356887] INFO: [Shinken] Checked 23 commands

[1424356887] INFO: [Shinken] Checking hostsextinfo...

[1424356887] INFO: [Shinken] Checked 0 hostsextinfo

[1424356887] INFO: [Shinken] Checking servicesextinfo...

[1424356887] INFO: [Shinken] Checked 0 servicesextinfo

[1424356887] INFO: [Shinken] Checking checkmodulations...

[1424356887] INFO: [Shinken] Checked 0 checkmodulations

[1424356887] INFO: [Shinken] Checking macromodulations...

[1424356887] INFO: [Shinken] Checked 0 macromodulations

[1424356887] INFO: [Shinken] Checking servicedependencies...

[1424356887] INFO: [Shinken] Checked 0 servicedependencies

[1424356887] INFO: [Shinken] Checking hostdependencies...

[1424356887] INFO: [Shinken] Checked 0 hostdependencies

[1424356887] INFO: [Shinken] Checking arbiters...

[1424356887] INFO: [Shinken] Checked 1 arbiters

[1424356887] INFO: [Shinken] Checking schedulers...

[1424356887] INFO: [Shinken] Checked 1 schedulers

[1424356887] INFO: [Shinken] Checking reactionners...

[1424356887] INFO: [Shinken] Checked 1 reactionners

[1424356887] INFO: [Shinken] Checking pollers...

[1424356887] INFO: [Shinken] Checked 1 pollers

[1424356887] INFO: [Shinken] Checking brokers...

[1424356887] INFO: [Shinken] Checked 1 brokers

[1424356887] INFO: [Shinken] Checking receivers...

[1424356887] INFO: [Shinken] Checked 1 receivers

[1424356887] INFO: [Shinken] Checking resultmodulations...

[1424356887] INFO: [Shinken] Checked 0 resultmodulations

[1424356887] INFO: [Shinken] Checking discoveryrules...

[1424356887] INFO: [Shinken] Checked 0 discoveryrules

[1424356887] INFO: [Shinken] Checking discoveryruns...

[1424356887] INFO: [Shinken] Checked 0 discoveryruns

[1424356887] INFO: [Shinken] Checking businessimpactmodulations...

[1424356887] INFO: [Shinken] Checked 0 businessimpactmodulations

[1424356887] INFO: [Shinken] Cutting the hosts and services into parts

[1424356887] INFO: [Shinken] Creating packs for realms

[1424356887] INFO: [Shinken] Number of hosts in the realm All: 1 (distributed in 1 linked packs)

[1424356887] INFO: [Shinken] Total number of hosts : 1

[1424356887] INFO: [Shinken] Things look okay - No serious problems were detected during the pre-flight check

[1424356887] INFO: [Shinken] [Arbiter] Serializing the configurations...

[1424356887] INFO: [Shinken] Using the default serialization pass

TOTAL serializing in 0.00950407981873

[1424356887] INFO: [Shinken] Configuration Loaded

valid pidfile exists (pid=31319) and not forced to replace. Exiting.

1. Quel est le port d'écoute TCP et quel est le processus à l'écoute ?

netstat -tlnp | grep "LISTEN .\*python"

tcp 0 0 0.0.0.0:7768 0.0.0.0:\* LISTEN 15657/python

Port TCP 7768 et processus numéro 15657.

1. Que fait la commande suivante ?

root@debianWheezy:~# service shinken check

Cette commande permet de vérifier les fichiers de configuration de shinken.

root@debianWheezy:~# service shinken check

Doing config check

. ok

Le résultat précédent montre que la configuration est syntaxiquement bonne.

On va introduire une erreur dans le fichier /etc/shinken/brokers/broker-master.cfg.

## Modules

# Default: None

# Interesting modules that can be used:

# - simple-log = just all logs into one file

# - livestatus = livestatus listener

# - tondodb-mysql = NDO DB support

# - npcdmod = Use the PNP addon

# - graphite = Use a Graphite time series DB for perfdata

# - webui = Shinken Web interface

# - glpidb = Save data in GLPI MySQL database

modules webUI

Et relancer le contrôle.

root@debianWheezy:~# service shinken check

Doing config check

[warn] full result is in /tmp/shinken\_checkconfig\_result ... (warning).

[FAIL] ConfigCheck failed: Configuration is incorrect, sorry, I bail out ... failed!

failed!

Cette fois ça ne marche pas, shinken nous indique d'aller consulter le fichier /tmp/shinken\_checkconfig\_result. On y trouve l'erreur, il ne reste plus qu'à rectifier.

[35m[1424357486] INFO: [Shinken] Checking brokers...[0m

[31m[1424357486] ERROR: [Shinken] [item::broker-master] Error: the module webUI is unknown for broker-master[0m

[31m[1424357486] ERROR: [Shinken] [items] In broker-master is incorrect ; from /etc/shinken/brokers/broker-master.cfg:15[0m

[31m[1424357486] ERROR: [Shinken] brokers conf incorrect!![0m

# Annexes

## Désinstaller manuellement Shinken

**Vérifier le statut de shinken et l'arrêter si nécessaire :**

root@debianWheezy:~# service shinken status

Checking status of scheduler

scheduler NOT RUNNING (pidfile (/var/run/shinken/schedulerd.pid) not exist)

failed!

Checking status of poller

poller NOT RUNNING (pidfile (/var/run/shinken/pollerd.pid) not exist)

failed!

Checking status of reactionner

reactionner NOT RUNNING (pidfile (/var/run/shinken/reactionnerd.pid) not exist)

failed!

Checking status of broker

broker NOT RUNNING (pidfile (/var/run/shinken/brokerd.pid) not exist)

failed!

Checking status of receiver

receiver NOT RUNNING (pidfile (/var/run/shinken/receiverd.pid) not exist)

failed!

Checking status of arbiter

arbiter NOT RUNNING (pidfile (/var/run/shinken/arbiterd.pid) not exist)

failed!

root@debianWheezy:~# ps aux | grep shinken

root 30005 0.0 0.0 7852 892 pts/0 S+ 11:54 0:00 grep shinken

**Supprimer l'utilisateur shinken :**

root@debianWheezy:~# deluser shinken

Suppression de l'utilisateur « shinken »...

Attention ! Le groupe « shinken » ne contient plus aucun membre.

Fait.

root@debianWheezy:~# delgroup shinken

Le groupe « shinken » n'existe pas.

**Supprimer tous les répertoires Shinken :**

root@debianWheezy:~# rm -RF /var/lib/shinken/

rm : option non valide -- F

Saisissez « rm --help » pour plus d'informations.

root@debianWheezy:~# rm -Rf /var/lib/shinken/

root@debianWheezy:~# rm -Rf /var/log/shinken/

root@debianWheezy:~# rm -Rf /var/mail/shinken/

root@debianWheezy:~# rm -Rf /var/home/shinken/

root@debianWheezy:~# rm -Rf /usr/bin/shinken

root@debianWheezy:~# rm -Rf /etc/shinken/

root@debianWheezy:~# rm -Rf /etc/default/shinken

root@debianWheezy:~# rm -Rf /etc/init.d/shinken

root@debianWheezy:~# rm -Rf /usr/local/lib/python2.7/dist-packages/shinken/

root@debianWheezy:~# rm -Rf /run/shinken/

root@debianWheezy:~# rm -Rf /usr/local/lib/python2.7/dist-packages/modules/

root@debianWheezy:~# rm -Rf /usr/local/lib/python2.7/dist-packages/Shinken-2.2.egg-info/

root@debianWheezy:~# find / -name shinken

root@debianWheezy:~#